



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/798,474	03/10/2004	Mark Vincent Scardina	50277-2389	7416
42425	7590	05/23/2008	EXAMINER	
HICKMAN PALERMO TRUONG & BECKER/ORACLE			TRAN, QUOC A	
2055 GATEWAY PLACE				
SUITE 550			ART UNIT	PAPER NUMBER
SAN JOSE, CA 95110-1083			2176	
			MAIL DATE	DELIVERY MODE
			05/23/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/798,474	SCARDINA ET AL.	
	Examiner	Art Unit	
	Quoc A. Tran	2176	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 13 February 2008.

2a) This action is **FINAL**. 2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-6, 13-25, 39-44 and 48-69 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) Claim(s) _____ is/are allowed.

6) Claim(s) 1-6, 13-25, 39-44 and 48-69 is/are rejected.

7) Claim(s) _____ is/are objected to.

8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on 13 February 2008 is/are: a) accepted or b) objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) All b) Some * c) None of:

1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. _____.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) Notice of References Cited (PTO-892)

2) Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5) Notice of Informal Patent Application

6) Other: _____.

Applicant's Response

This action is a **Final** rejection in response to Applicant's Amendment/Remarks filed on 02/13/2008.

Applicant argued against the rejections to claims 1-6, 13-25, 39-44 and 48-69 under 103 (a) previously set forth in the Office Action dated 05/31/2007.

Claims 1, 3, 13, 23, 39, 42-44, 48, 50, 54, 64, and 67-69 have been amended. Claims 7-12, 26-38 and 45-47 were previously cancelled. Claims 1-6, 13-25, 39-44, and 48-69 are pending in the present application. Effective filing date is 03/10/2004, priority date **09/04/2003** (Assignee Oracle).

It is noted claims 1-6, 42-44 and 48-69 rejected under 112 second paragraph previously presented in the Office Action dated 11/13/2007, has been withdrawn due to applicant's amendments filed 02/13/2008.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-6,13-25, 39-44, and 48-69, are rejected under 35 U.S.C. 103(a) as being unpatentable over Fry US 20030159112A1, filed 11/26/2002 (hereinafter Fry, in view of in view of Sijacic et al. US 20020184145A1 filed 05/31/2001 (hereinafter Sijacic).

Independent claim 1,

Fry teaches:

A method comprising the computer-implemented steps of: while an XML processor performs a validation operation on an XML-based input stream, wherein the XML processor is configured to send validated XML data to an application,

(See Para 22-27→ Fry disclose SAX as a streaming parser (i.e. XML validator in streaming fashion, wherein the data being validated while streaming.)

performing the steps of: while validating a particular XML element in said XML-based input stream,

(See Para 22-27 → Fry disclose SAX as a streaming parser (i.e. XML validator in streaming fashion, wherein the data being validated while streaming.)

In addition, Fry does not explicitly teach, but Sijacic teaches:

causing said XML processor to generate one or more messages that indicate to the application how the application is to process said particular XML element, by identifying one or more annotations that are associated with said particular XML element;

(See Fig. 5-6 and the Abstract and Para 11, 32, 43 and 58-62 → Sijacic discloses this limitation in that fig. 5-6 shown steps 610-660 (i.e. validating request, parsing, reposes to message) wherein the request messages are parsed by event-based parser or API such as Simple API for XML (SAX) a parser operating within the XML DOM 510 (Step 620 Fig. 6), and further defines the logical structure of these documents and the manner by which they are edited and accessed. This structure, or model, enables XML servlet 222 to identify interfaces and objects used to represent and modify a document; the behavior and attributes of these interfaces and objects; and any relationships between the interfaces and object.)

and responding to a request for information about said particular XML element by providing said one or more messages.

(See Fig. 5-6 and the Abstract and Para 11, 32, 43 and 58-62 → Sijacic discloses this limitation in that fig. 5-6 shown steps 610-660 (i.e. validating request, parsing, reposes

to message) wherein the request messages are parsed by event-based parser or API such as Simple API for XML (SAX) a parser operating within the XML DOM 510 (Step 620 Fig. 6), and further defines the logical structure of these documents and the manner by which they are edited and accessed. This structure, or model, enables XML servlet 222 to identify interfaces and objects used to represent and modify a document; the behavior and attributes of these interfaces and objects; and any relationships between the interfaces and object, thus enabling object-oriented programming to be used to complete conversions to XML formats implemented by biller manager 144 of EIPP (electronic invoice presentment and payment) system.)

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML SAX Streaming parser API to include a means of said causing said XML processor to generate one or more messages that indicate to the application how the application is to process said particular XML element, by identifying one or more annotations that are associated with said particular XML element as taught by Sijacic, because Fry and Sijacic are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and provides a predictable result of allows the a SAX or DOM parser includes name of a selected element to be passed to the method. The base parser can begin processing the XML document to locate an element tag signifying an element of the XML document. The iterative method can then direct the base parser to step through the elements in the document until the

tag is located that corresponds to the selected element. The base parser can extract the selected element from the XML document and process the element such as by generating an event that can be read by a Java application. The event can then be placed on an event stream for use by an application- See Fry at Para 17.)

Independent claim 13,

is fully incorporated similar subject of claim 1 cited above, and are similarly rejected along the same rationale. Thus, Fry and Sijacic disclose every limitation of Claim 13 and provide proper reasons to combine, as indicated in the above rejections for Claim 1. In addition, Fry teaches:

particular information comprises one or more of: the name, the data type of the node currently being process stream,

(See Para 17, 27 → Fry disclose a SAX or DOM parser, which allows the name, type of a selected element to be passed.)

current validation mode is one of strict mode, lax mode, and skip mode;

(See Fry at the Abstract → disclose this limitation in that the XML processor selects and instantiates a streaming parser API, then provides a variety of types of access to the application or client that does not require the entire document to be read into memory, including providing an XML stream, pulling XML information, and skipping unwanted XML from the document (i.e. validating and skipping)).

Independent claim 39,

Claim 32 recites a computer -readable medium store instruction to implement a method recited in Claims 1 and 13. Thus, Fry and Sijacic disclose every limitation of Claim 39 and provide proper reasons to combine, as indicated in the above rejections for Claims 1 and 13, see also Fry at Para 10, discloses memory (i.e. computer-readable medium.)

In addition, Fry teaches:

a validator that validates elements and attributes in an XML-based input stream against information that dictates the structure of corresponding elements and attributes,

(See Para 22-27 → Fry disclose this limitation in that the Streaming parser API, is utilized for validating, parsing XML document. Such a streaming parser for XML can be implemented on top of SAX. The streaming parser takes SAX events and constructs an easily manipulated event stream that is available to the application programmer. The streaming parser gives parsing control to the programmer by exposing a simple iteration-based API to the programmer.)

wherein said particular information about validating said first element comprises one or more of: the name of said first element; the data type of said first element;

(See Para 17, 27 → Fry disclose a SAX or DOM parser, which allows the name, type of a selected element to be passed.)

In addition, Fry does not explicitly teach, but Sijacic teaches:

**a state machine that responds to requests for particular information,
...performance on said first element.**

(See Para 34 and 58-62 → Sijacic disclose this limitation in that process manager 142 is a manages data that pertains to the current state of items in a given workflow process utilized XML DOM 510, defines the logical structure of these documents and the manner by which they are edited and accessed. This structure, or model, enables XML servlet 222 to identify interfaces and objects used to represent and modify a document; the behavior and attributes of these interfaces and objects; and any relationships between the interfaces and object.)

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML SAX Streaming parser API to include a means of said a state machine that responds to requests for particular information, ...performance on said first element as taught by Sijacic, because Fry and Sijacic are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and provides a predictable result of allows the a SAX or DOM parser includes name of a selected element to be passed to the method. The base parser can begin processing the XML document to locate an element tag signifying an element of the XML document. The iterative method can then direct the base parser to step through the elements in the document until the tag is located that corresponds to the selected element. The base

parser can extract the selected element from the XML document and process the element such as by generating an event that can be read by a Java application. The event can then be placed on an event stream for use by an application- See Fry at Para 17.)

Independent Claim 48:

Claim 48 recites a computer-readable storage medium store instruction to implement a method recited in Claim 1. Thus, Fry and Sijacic disclose every limitation of Claim 48 and provide proper reasons to combine, as indicated in the above rejections for Claim 1, see also Fry at Para 10, discloses memory (i.e. computer-readable medium.)

Independent Claim 54:

Claim 54 recites a computer-readable storage medium store instruction to implement a method recited in Claim 13. Thus, Fry and Sijacic disclose every limitation of Claim 54 and provide proper reasons to combine, as indicated in the above rejections for Claim 13, see also Fry at Para 10, discloses memory (i.e. computer-readable medium.)

Claim 2,

Fry and Sijacic teach the method of claim 1 and further comprise:

while said XML processor performs said validation operation on said XML-based input stream.

(See Para 22-27 → Fry disclose this limitation in that the Streaming parser API is utilized for validating, parsing XML document. Such a streaming parser for XML can be implemented on top of SAX. The streaming parser takes SAX events and constructs an easily manipulated event stream that is available to the application programmer. The streaming parser gives parsing control to the programmer by exposing a simple iteration-based API to the programmer.)

In addition, Fry does not explicitly teach, but Sijacic teaches:

receiving a request for said one or more annotations; wherein the step of causing said XML processor to generate one or more messages is performed in response to said request.

(See Fig.5-6 and the Abstract and Para 11, 32, 43 and Para 62 → Sijacic discloses this limitation in that XML DOM, XML Parser, event-based parser or API such as Simple API for XML (SAX). This process allows XML servlet 222 to recognize how the messages are represented as objects. The XML servlet validates the request message to ensure the request message is in XML format and includes a transformer tag that designates the type of corresponding response message required that is conformed to a particular document type definition (XML element), thus enabling object-oriented programming to

be used to complete conversions to XML formats implemented by biller manager 144 of EIPP (electronic invoice presentment and payment) system.)

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML SAX Streaming parser API to include a means of said receiving a request for said one or more annotations; wherein the step of causing said XML processor to generate one or more messages is performed in response to said request as taught by Sijacic, because Fry and Sijacic are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and provides a predictable result of allows the a SAX or DOM parser includes name of a selected element to be passed to the method. The base parser can begin processing the XML document to locate an element tag signifying an element of the XML document. The iterative method can then direct the base parser to step through the elements in the document until the tag is located that corresponds to the selected element. The base parser can extract the selected element from the XML document and process the element such as by generating an event that can be read by a Java application. The event can then be placed on an event stream for use by an application- See Fry at Para 17.)

Claim 3,

Fry and Sijacic teach the method of claim 2 and further comprise:

wherein the step of receiving a request includes receiving a request via an application program interface through which information about said validation operation can be requested by an the application.

(See Para 8 --> Fry described the eXtensible Markup Language (XML) has become a standard for inter-application communication wherein XML messages passing between applications contain tags with self-describing text. The self-describing text allows these messages to be understandable not only to the applications, but also to humans reading an XML document.

Also see Fig. 1 and Para 22-27, 32 and 37 → Fry further the XML processing, forming the base class for all XML processors in the parsing paradigm, including for example the StreamParser and SAXDriver (generating SAX events and implement an XMLReader class from SAX see Para 37). The base parser iterates over XML Elements, which can then be encapsulated in the Element class that enforcing higher-level well-formedness constraints, such as proper element nesting and proper namespace declaration)

Claim 4,

Fry and Sijacic teach the method of claim 1 and further comprise:

wherein the step of causing said XML processor to generate one or more messages includes causing said XML processor to generate one or more messages that are transmitted in an output stream.

(See Fig.5-6 and the Abstract and Para 11, 32, 43 and Para 62 →Sijacic discloses this limitation in that XML DOM, XML Parser, event-based parser or API such as Simple API for XML (SAX).This process allows XML servlet 222 to recognize how the messages are represented as objects .The XML servlet validates the request message to ensures the request message is in XML format and includes a transformer tag that designates the type of corresponding response message required that is conformed to a particular document type definition (XML element), thus enabling object-oriented programming to be used to complete conversions to XML formats implemented by biller manager 144 of EIPP (electronic invoice presentment and payment) system.)

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML SAX Streaming parser API to include a means of said causing said XML processor to generate one or more messages includes causing said XML processor to generate one or more messages that are transmitted in an output stream as taught by Sijacic, because Fry and Sijacic are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and provides a predictable result of allows the a SAX or DOM parser includes name of a selected element to be

passed to the method. The base parser can begin processing the XML document to locate an element tag signifying an element of the XML document. The iterative method can then direct the base parser to step through the elements in the document until the tag is located that corresponds to the selected element. The base parser can extract the selected element from the XML document and process the element such as by generating an event that can be read by a Java application. The event can then be placed on an event stream for use by an application- See Fry at Para 17.)

Claim 5,

Fry and Sijacic teach the method of claim 1 and further comprise:

wherein the step of causing said XML processor to generate one or more messages includes causing said XML processor to generate one or more messages before completion of said validation operation on said XML-based input stream,

(See Fig.5-6 and the Abstract and Para 11, 32, 43 and Para 62 →Sijacic discloses this limitation in that XML DOM, XML Parser, event-based parser or API such as Simple API for XML (SAX).This process allows XML servlet 222 to recognize how the messages are represented as objects .The XML servlet validates the request message to ensures the request message is in XML format and includes a transformer tag that designates the type of corresponding response message required that is conformed to a particular document type definition (XML element), thus enabling object-oriented programming to

be used to complete conversions to XML formats implemented by biller manager 144 of EIPP (electronic invoice presentment and payment) system.)

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML SAX Streaming parser API to include a means of said causing said XML processor to generate one or more messages includes causing said XML processor to generate one or more messages before completion of said validation operation on said XML-based input stream as taught by Sijacic, because Fry and Sijacic are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and provides a predictable result of allows the a SAX or DOM parser includes name of a selected element to be passed to the method. The base parser can begin processing the XML document to locate an element tag signifying an element of the XML document. The iterative method can then direct the base parser to step through the elements in the document until the tag is located that corresponds to the selected element. The base parser can extract the selected element from the XML document and process the element such as by generating an event that can be read by a Java application. The event can then be placed on an event stream for use by an application- See Fry at Para 17.)

Claim 6,

Fry and Sijacic teach the method of claim 1 and further comprise:

wherein said validation operation includes performing a validation operation on said particular XML element of said XML-based input stream;

(See Para 22-27 → Fry disclose SAX as a streaming parser (i.e. XML validator in streaming fashion, wherein the data being validated while streaming.)

In addition, Fry does not explicitly teach, but Sijacic teaches:

causing said XML processor to generate one or more messages includes causing said XML processor to generate one or more messages that indicate how to process said particular XML element, only if said particular XML element is determined valid based on said validation operation on said particular XML element,

(See Fig.5-6 and the Abstract and Para 11, 32, 43 and Para 62 → Sijacic discloses this limitation in that XML DOM, XML Parser, event-based parser or API such as Simple API for XML (SAX). This process allows XML servlet 222 to recognize how the messages are represented as objects. The XML servlet validates the request message to ensure the request message is in XML format and includes a transformer tag that designates the type of corresponding response message required that is conformed to a particular document type definition (XML element), thus enabling object-oriented programming to be used to complete conversions to XML formats implemented by biller manager 144 of EIPP (electronic invoice presentment and payment) system.)

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML SAX Streaming parser API to include a means of said causing said XML processor to generate one or more messages includes causing said XML processor to generate one or more messages that indicate how to process said particular XML element, only if said particular XML element is determined valid based on said validation operation on said particular XML element as taught by Sijacic, because Fry and Sijacic are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and provides a predictable result of allows the a SAX or DOM parser includes name of a selected element to be passed to the method. The base parser can begin processing the XML document to locate an element tag signifying an element of the XML document. The iterative method can then direct the base parser to step through the elements in the document until the tag is located that corresponds to the selected element. The base parser can extract the selected element from the XML document and process the element such as by generating an event that can be read by a Java application. The event can then be placed on an event stream for use by an application- See Fry at Para 17.)

Claims 14-16,

Claims 14-16 recite a computer-readable storage medium store instruction to implement a method recited in Claim 39. Thus, Fry and Sijacic disclose every limitation of Claims 14-16 and provide proper reasons to combine, as indicated

in the above rejections for Claim 13, see also Fry at Para 10, discloses memory (i.e. computer-readable medium.)

Claim 17,

Fry and Sijacic teach the method of claim 13 and further comprise:

wherein the step of receiving a request includes receiving a request regarding whether a data type of content of first element of said XML-based input stream conforms to a corresponding data type definition in information that dictates the structure of corresponding XML data;

(See Para 20→ Fry disclose this limitation in that the system and method of Fry supported three levels of processing: raw, non-validating, and validating with XML schemas or Document Type Definitions (DTDs). These parsers can be selected by instantiating different implementations of a streaming XML API. A pull-parser streaming API can also support data binding implementations SAX as a streaming parser (i.e. XML validator in streaming fashion, wherein the data is being validated while streaming.)

Claim 18,

Fry and Sijacic teach the method of claim 13 and further comprise:

wherein the step of receiving a request includes receiving a request regarding a first annotation that is associated with first element of said XML-based input stream,

(See Para 22-27 → Fry disclose SAX as a streaming parser (i.e. XML validator in streaming fashion, wherein the data being validated while streaming.)

In addition, Fry does not explicitly teach, but Sijacic teaches:

**wherein said first annotation is defined in information that dictates
the structure of corresponding XML data;**

(See Para 34 and 58-62 → Sijacic disclose this limitation in that process manager 142 is a manages data that pertains to the current state of items in a given workflow process utilized XML DOM 510, defines the logical structure of these documents and the manner by which they are edited and accessed. This structure, or model, enables XML servlet 222 to identify interfaces and objects used to represent and modify a document; the behavior and attributes of these interfaces and objects; and any relationships between the interfaces and object.)

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML SAX Streaming parser API to include a means of said first annotation is defined in information that dictates the structure of corresponding XML data as taught by Sijacic, because Fry and Sijacic are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and provides a predictable result of allows the a SAX or DOM parser includes name of a selected element to be passed to the method. The base parser can begin processing the XML document to

locate an element tag signifying an element of the XML document. The iterative method can then direct the base parser to step through the elements in the document until the tag is located that corresponds to the selected element. The base parser can extract the selected element from the XML document and process the element such as by generating an event that can be read by a Java application. The event can then be placed on an event stream for use by an application- See Fry at Para 17.)

Claim 19,

Fry and Sijacic teach the method of claim 18 and further comprise:

**wherein the step stream is validated in said validation operation,
comprises a plurality of schema definitions that are associated with a
plurality of corresponding XML documents that could be constituent to
said XML-based input stream.**

(See Para 22-27 → Fry disclose SAX as a streaming parser (i.e. XML validator in streaming fashion, wherein the data being validated while streaming. Also Fry further discloses at Para 20, the system and method of Fry supported three levels of processing: raw, non-validating, and validating with XML schemas or Document Type Definitions (DTDs). These parsers can be selected by instantiating different implementations of a streaming XML API. A pull-parser streaming API can also support data binding implementations SAX as a streaming parser (i.e. XML validator in streaming fashion, wherein the data is being validated while streaming.)

In addition, Fry does not explicitly teach, but Sijacic teaches:

wherein said information that dictates the structure of corresponding XML data comprises a second annotation definition that is associated with a second element of said XML-based input stream,

(See Para 34 and 58-62 → Sijacic disclose this limitation in that process manager 142 is a manages data that pertains to the current state of items in a given workflow process utilized XML DOM 510, defines the logical structure of these documents and the manner by which they are edited and accessed. This structure, or model, enables XML servlet 222 to identify interfaces and objects used to represent and modify a document; the behavior and attributes of these interfaces and objects; and any relationships between the interfaces and object.)

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML SAX Streaming parser API to include a means of said information that dictates the structure of corresponding XML data comprises a second annotation definition that is associated with a second element of said XML-based input stream as taught by Sijacic, because Fry and Sijacic are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and provides a predictable result of allows the a SAX or DOM parser includes name of a selected element to be passed to the method. The base parser can begin processing the XML document to locate an element tag signifying an element of the XML document. The iterative method

can then direct the base parser to step through the elements in the document until the tag is located that corresponds to the selected element. The base parser can extract the selected element from the XML document and process the element such as by generating an event that can be read by a Java application. The event can then be placed on an event stream for use by an application- See Fry at Para 17.)

Claim 20,

Fry and Sijacic teach the method of claim 13 and further comprise:

wherein the step of receiving a request includes receiving a request regarding a status of said validation operation with respect to a first element of said XML-based input stream.

(See Para 34 and 58-62 → Sijacic disclose this limitation in that process manager 142 is a manages data that pertains to the current state of items in a given workflow process utilized XML DOM 510, defines the logical structure of these documents and the manner by which they are edited and accessed. This structure, or model, enables XML servlet 222 to identify interfaces and objects used to represent and modify a document; the behavior and attributes of these interfaces and objects; and any relationships between the interfaces and object.)

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML SAX Streaming parser API to include a means of said receiving a request includes receiving a request regarding a

status of said validation operation with respect to a first element of said XML-based input stream as taught by Sijacic, because Fry and Sijacic are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and provides a predictable result of allows the a SAX or DOM parser includes name of a selected element to be passed to the method. The base parser can begin processing the XML document to locate an element tag signifying an element of the XML document. The iterative method can then direct the base parser to step through the elements in the document until the tag is located that corresponds to the selected element. The base parser can extract the selected element from the XML document and process the element such as by generating an event that can be read by a Java application. The event can then be placed on an event stream for use by an application- See Fry at Para 17.)

Claim 21,

Fry and Sijacic teach the method of claim 13 and further comprise:

wherein the step of receiving a request includes receiving a request via an application program interface through which information about said validation operation can be requested by an external application.

(See Fig. 1 and Para 22-27→ Fry disclose SAX as a streaming parser (i.e. XML validator in streaming fashion, wherein the data being validated while streaming whereby the client application and/or XML document can be external (See Fig. 1 item 100 and 102).)

Claim 22,

Fry and Sijacic teach the method of claim 13 and further comprise:

wherein the step of receiving a request includes receiving a request from an event handler sent in response to an event received in a parser output stream.

(See Para 22-27 → Fry disclose SAX as a streaming parser (i.e. XML validator in streaming fashion, wherein the data being validated while streaming. Also Fry further discloses at Para 25, the streaming API can provide an efficient way to handle XML documents that is uniquely suited to the runtime needs of an application server. A streaming API can be implemented on top of an existing XML parser. This approach can also be referred to as "pull-parsing" or "event-based processing.")

Claim 23,

Fry and Sijacic teach the method of claim 13 and further comprise:

wherein the step of responding to said request includes providing, in an output stream, said particular information about the state of said validation operation.

(See Para 34 and 58-62 → Sijacic disclose this limitation in that process manager 142 is a manages data that pertains to the current state of items in a given workflow process utilized XML DOM 510, defines the logical structure of these documents and the manner by which they are edited and accessed. This structure, or model, enables XML servlet 222 to identify interfaces and objects used to represent and modify a document;

the behavior and attributes of these interfaces and objects; and any relationships between the interfaces and object.)

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML SAX Streaming parser API to include a means of said responding to said request includes providing, in an output stream, said particular information about the state of said validation operation as taught by Sijacic, because Fry and Sijacic are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and provides a predictable result of allows the a SAX or DOM parser includes name of a selected element to be passed to the method. The base parser can begin processing the XML document to locate an element tag signifying an element of the XML document. The iterative method can then direct the base parser to step through the elements in the document until the tag is located that corresponds to the selected element. The base parser can extract the selected element from the XML document and process the element such as by generating an event that can be read by a Java application. The event can then be placed on an event stream for use by an application- See Fry at Para 17.)

Claim 24,

Fry and Sijacic teach the method of claim 13 and further comprise:

parsing said XML-based input stream only once for both of said validation operation, and operations that are dictated by annotations associated with elements in said XML-based input stream.

(See Para 22-27 → Fry disclose SAX as a streaming parser (i.e. XML validator in streaming fashion, wherein the data being validated while streaming.)

Claim 25,

Fry and Sijacic teach the method of claim 13 and further comprise:

wherein information that dictates the structure of corresponding XML data in said XML-based input stream, with which said input stream is validated in said validation operation, comprises a plurality of schema definitions that are associated with a plurality of corresponding XML documents that could be constituent to said XML-based input stream.

(See Para 34 and 58-62 → Sijacic disclose this limitation in that process manager 142 is a manages data that pertains to the current state of items in a given workflow process utilized XML DOM 510, defines the logical structure of these documents and the manner by which they are edited and accessed. This structure, or model, enables XML servlet 222 to identify interfaces and objects used to represent and modify a document; the behavior and attributes of these interfaces and objects; and any relationships between the interfaces and object.)

Claim 40,

Claim 40 recite a computer-readable storage medium store instruction to implement a method recited in Claims 13-15. Thus, Fry and Sijacic disclose every limitation of Claim 40 and provide proper reasons to combine, as indicated in the above rejections for Claims 13-15, see also Fry at Para 10, discloses memory (i.e. computer-readable medium.)

Claim 41,

Claim 41 recites a computer-readable storage medium store instruction to implement a method recited in Claims 13 and 22. Thus, Fry and Sijacic disclose every limitation of Claim 41 and provide proper reasons to combine, as indicated in the above rejections for Claims 13 and 22, see also Fry at Para 10, discloses memory (i.e. computer-readable medium.)

Claims 42-43,

Fry and Sijacic teach the method of claim 1 and further comprise:

**reading said one or more metadata, XML schema that corresponds to
said XML-based input stream.**

(See Para 22-27 → Fry disclose SAX as a streaming parser (i.e. XML validator in streaming fashion, wherein the data being validated while streaming.

In addition, Fry does not explicitly teach, but Sijacic teaches:

**reading said one or more annotation from metadata, XML schema
that corresponds to said XML-based input stream that corresponds to said
XML based input stream,**

(See the Abstract and Para 34 and 58-62 → Sijacic disclose this limitation in that the servlet ensures the request message is in XML format and includes a transformer tag that designates the type of corresponding response message required. The XML servlet validates the request message to ensure it conforms to a particular document type definition (it is reasonably interprets the XML data as meta data and its schema is designates the type of corresponding response message required of the XML data that is utilized process manager 142 is a manages data that pertains to the current state of items in a given workflow process utilized XML DOM 510 (i.e. streaming) defines the logical structure of these documents and the manner by which they are edited and accessed. This structure, or model, enables XML servlet 222 to identify interfaces and objects used to represent and modify a document; the behavior and attributes of these interfaces and objects; and any relationships between the interfaces and object.)

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML SAX Streaming parser API to include a means of said reading said one or more annotation from metadata, XML schema that corresponds to said XML-based input stream that corresponds to said XML based input stream as taught by Sijacic, because Fry and Sijacic are analogous art,

since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and provides a predictable result of allows the a SAX or DOM parser includes name of a selected element to be passed to the method. The base parser can begin processing the XML document to locate an element tag signifying an element of the XML document. The iterative method can then direct the base parser to step through the elements in the document until the tag is located that corresponds to the selected element. The base parser can extract the selected element from the XML document and process the element such as by generating an event that can be read by a Java application. The event can then be placed on an event stream for use by an application- See Fry at Para 17.)

(See Para 34 and 58-62 → Sijacic disclose this limitation in that process manager 142 is a manages data that pertains to the current state of items in a given workflow process utilized XML DOM 510, defines the logical structure of these documents and the manner by which they are edited and accessed. This structure, or model, enables XML servlet 222 to identify interfaces and objects used to represent and modify a document; the behavior and attributes of these interfaces and objects; and any relationships between the interfaces and object.)

Claim 44,

Fry and Sijacic teach the method of claim 1 and further comprise:

wherein the step of causing said XML processor to generate one or more messages includes causing said XML processor to generate one or more messages that indicate to the application how to conform said particular XML element to one or more requirements of the application that uses said specific particular XML element.

(See the Abstract and Para 22-27 → Fry disclose this limitation in that an application or client needing access to an XML document can contact an XML parser, XML processor, or XML reader in order to gain access to the document. The XML processor selects and instantiates a streaming parser API that is appropriate for the XML document and the client or application. Streaming parser APIs include SAX parser as a streaming parser (i.e. XML validator in streaming fashion.)

Claims 49-53 respectively:

Claims 49-53 recite a computer-readable storage medium store instruction to implement a method recited in Claims 2-6. Thus, Fry and Sijacic disclose every limitation of Claims 49-53 respectively and provide proper reasons to combine, as indicated in the above rejections for Claims 13 and 22, see also Fry at Para 10, discloses memory (i.e. computer-readable medium.)

Claims 55-69 respectively

Claim 55-69 recite a computer-readable storage medium store instruction to implement a method recited in Claims 14-25 and 42-44 respectively. Thus, Fry and Sijacic disclose every limitation of Claims 55-69 respectively and provide proper reasons to combine, as indicated in the above rejections for Claims 13 and 22, see also Fry at Para 10, discloses memory (i.e. computer-readable medium.)

It is noted that any citations to specific, pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains and may be relied upon for all that it would have reasonably suggested to one having ordinary skill in the art.

See, MPEP 2123.

Response to Arguments

The Remarks filed on 08/31/2007 has been fully considered but are moot but in view of the new ground(s) of rejection. It is noted applicant's amendment necessitated the new ground(s) of rejection presented in this Office action (see rejection for details).

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Quoc A. Tran whose telephone number is 571-272-8664. The examiner can normally be reached on Monday through Friday from 9 AM to 5 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Doug Hutton can be reached on (571)272-4137. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Quoc A. Tran/
Examiner, Art Unit 2176

/Rachna S Desai/
Primary Examiner, Art Unit 2176